

ESAME DI ALGORITMI E STRUTTURE DATI
A.A. 2014/2015 – APPELLO DEL 01/09/2015

TEMPO A DISPOSIZIONE: 120 MINUTI

ESERCIZIO 1:

1. Scrivere un algoritmo

`isMinHeap(array A) → boolean`

il quale, dato un array di interi **A** di dimensione **n**, termina restituendo *true* se **A** rappresenta un **MIN-HEAP** e *false* altrimenti.

2. Calcolare la complessità nel caso peggiore e nel caso migliore dell'algoritmo proposto, motivando le risposte.

ESERCIZIO 2:

- i) Si scriva lo pseudocodice dell'algoritmo `BubbleSort` in grado di ordinare gli elementi contenuti in un array **A** di **n** interi preso in ingresso.
- ii) Descrivere i passi eseguiti dall'algoritmo `BubbleSort` sul seguente array in input.

0	1	2	3	4	5	6	7
8	1	5	4	3	9	2	6

- iii) Indicare la complessità dell'algoritmo `BubbleSort` nel caso peggiore.

ESERCIZIO 3:

Sia dato l'insieme di chiavi $\mathbf{K} = \{ 18, 20, 66, 54, 26, 16, 64, 98, 46 \}$ e sia $\mathbf{m} = 10$.

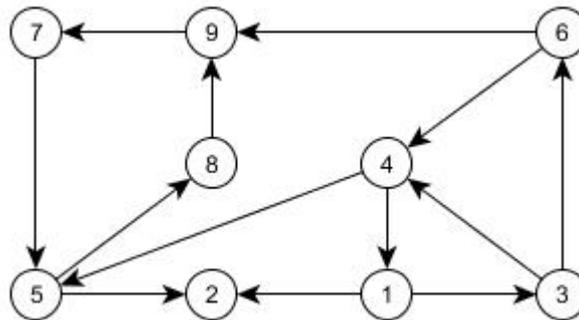
1. Calcolare per ogni chiave \mathbf{k} di \mathbf{K} la funzione di hash

$$h(k) = (k/2) \bmod m$$

2. Dire, motivando la risposta, se la funzione di hash h suindicata è o non è una funzione di hash perfetta.
3. Inserire le chiavi dell'insieme \mathbf{K} in una tabella hash (inizialmente vuota) di dimensione \mathbf{m} basata su *liste di collisione*.

ESERCIZIO 4:

- i) Si scriva lo pseudocodice dell'algoritmo `visitaBFS(vertice v)` → albero che esegue la visita in ampiezza di un grafo orientato.
- ii) Descrivere i passi dell'algoritmo di visita quando applicato al grafo orientato rappresentato in figura partendo dal nodo **1** e presentare l'albero BFS prodotto dalla visita.



ESERCIZIO 5:

Indicare i vantaggi e gli svantaggi della rappresentazione dei dati indicizzata (es. tramite array) e della rappresentazione dei dati collegata (es. liste).